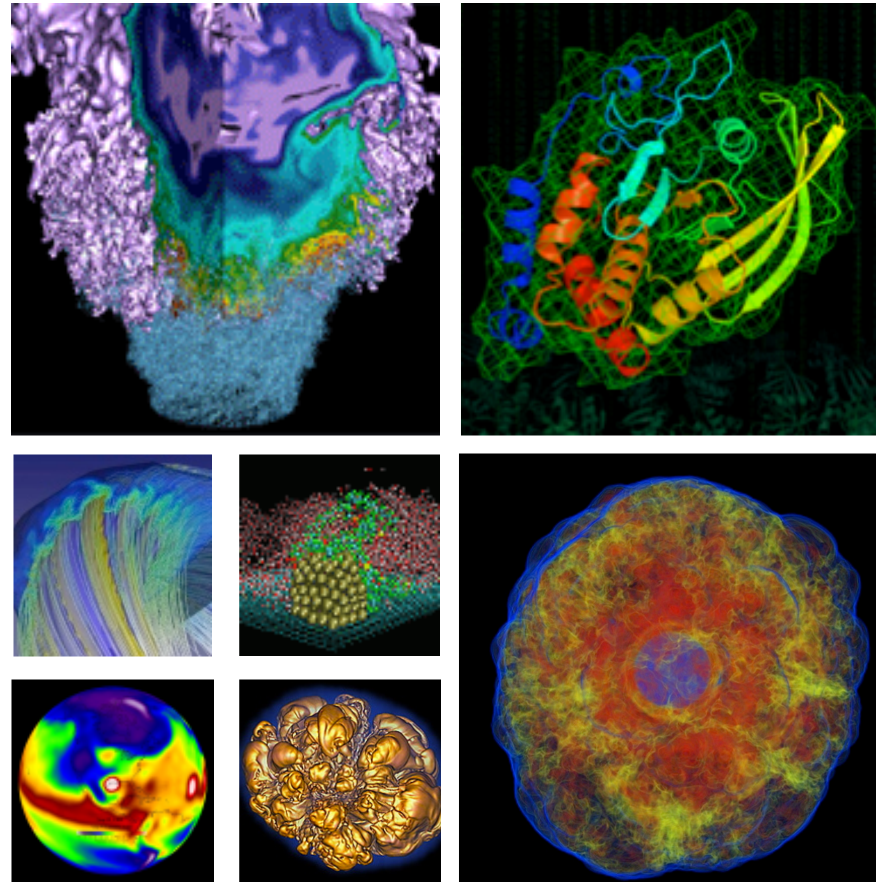
Accelerating Science with the NERSC Burst Buffer



Debbie Bard

Big Data Architect,
Data and Analytics Services
NERSC, LBL

July 22, 2016

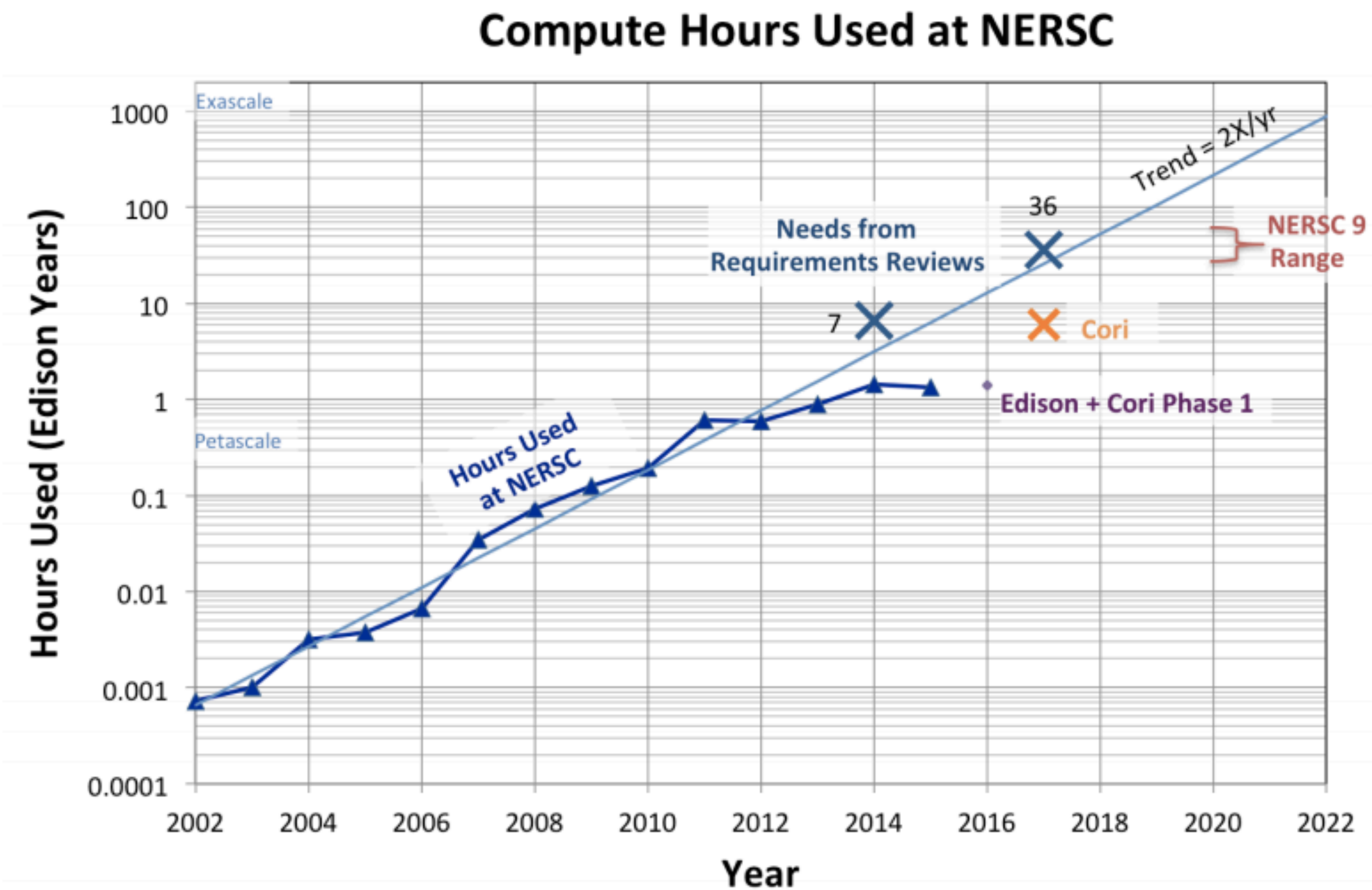
U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB

Outline

- **Future computing architecture**
  – The New Storage Hierarchy
- **What is a Burst Buffer?**
  – Architecture and software
- **Users are excited about new architectures!**
  – Early User Program
- **Science applications ≠ benchmarks**
  – Real-world performance
- **New tech teething problems**
  – Challenges and Lessons Learned

Our users are demanding…



… and not just for more compute time!

- **Users biggest "ask" (after wanting more compute cycles) is for better IO performance**
  - Eg scale up a simulation from 100k cores to 1M cores – 10x more compute producing 10x more data *per timestep*. Need 10x more IO BW!
  - Memory can be the largest dollar and power cost in an HPC system
- **New chip architectures (eg Knight's Landing) are very energy efficient – provide the required compute for less power**
  - But to use them well, you have to be able to corral your data appropriately

HPC memory hierarchy is changing

Past (Edison)

On Chip

CPU

Memory (DRAM)

Off Chip

Storage (HDD)

Present (Cori)

On Chip

CPU

Near Memory (HBM)

Far Memory (DRAM)

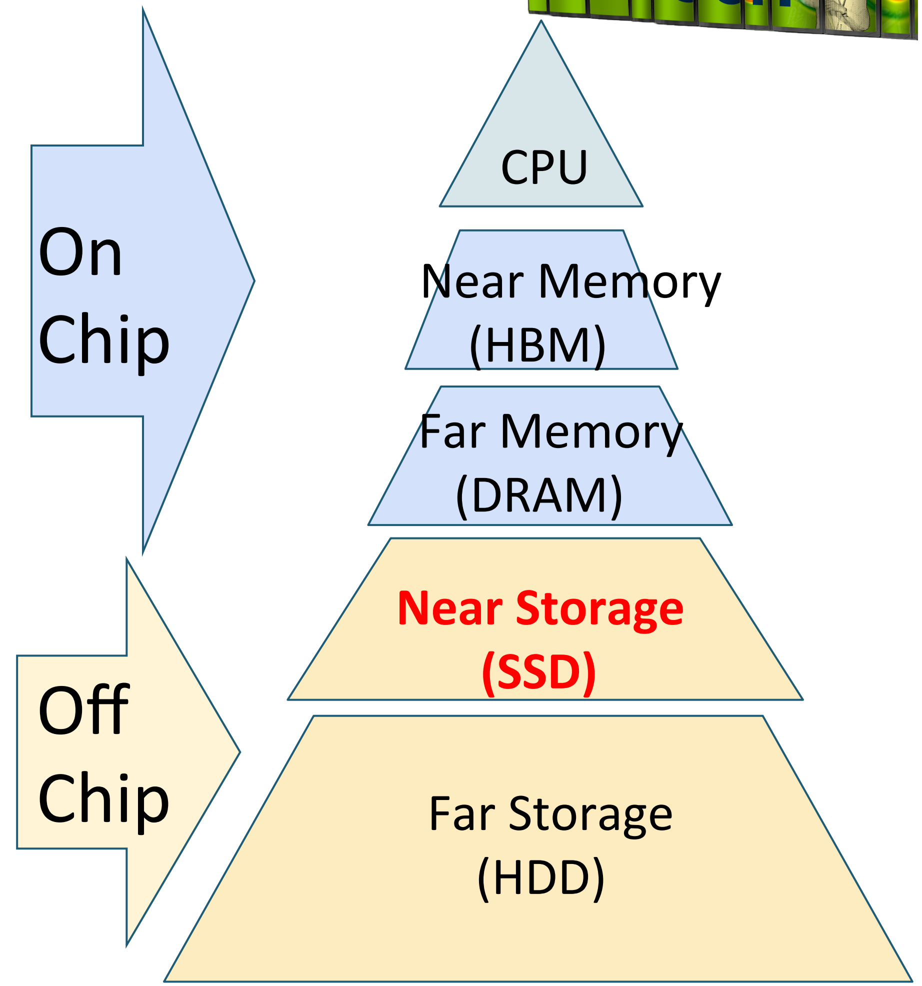
Off Chip

Near Storage (SSD)

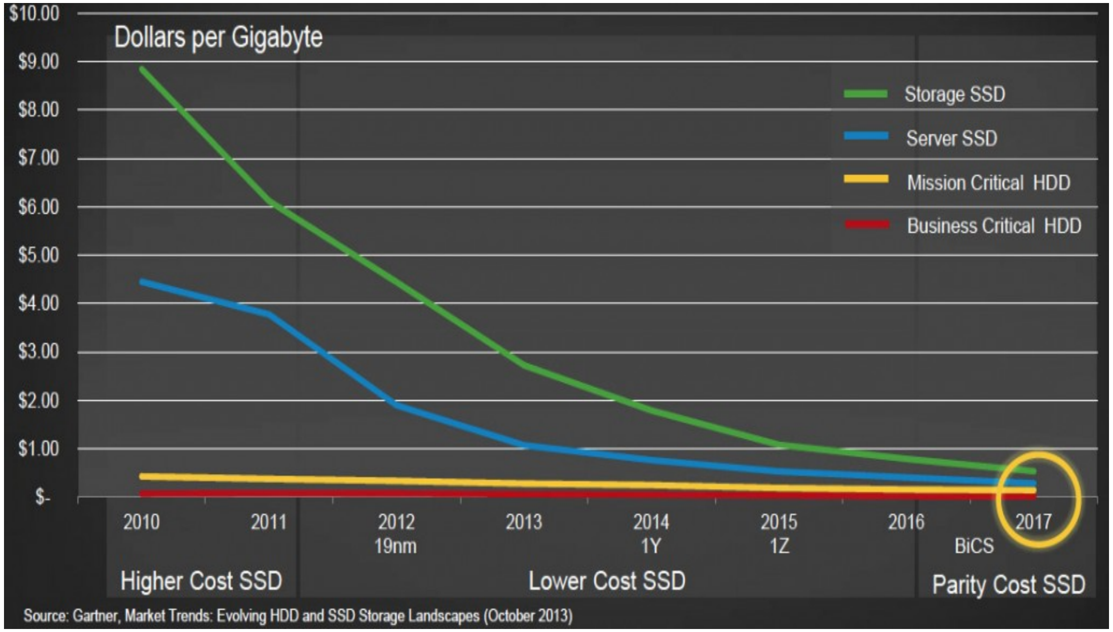
Far Storage (HDD)

HPC memory hierarchy is changing

- ***Silicon and system integration***
- **Bring everything – storage, memory, interconnect – closer to the cores**
- **Raise center of gravity of memory pyramid, and make it fatter**
  - *Enable faster and more efficient data movement*

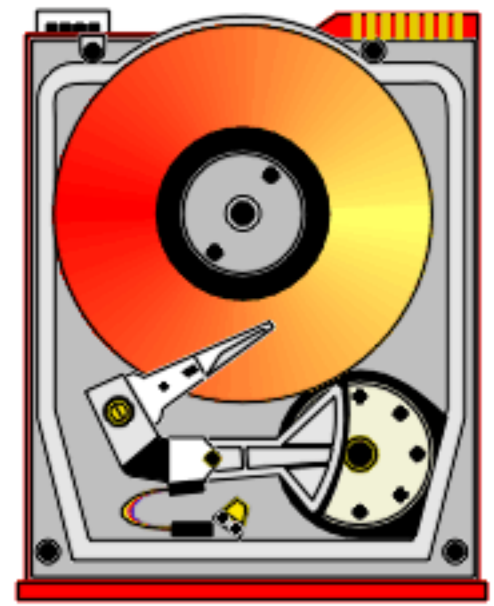


- **HDD capacity/$ is increasing over time, but SSD is catching up fast!**
- **BW and IOPs are flat for HDD**



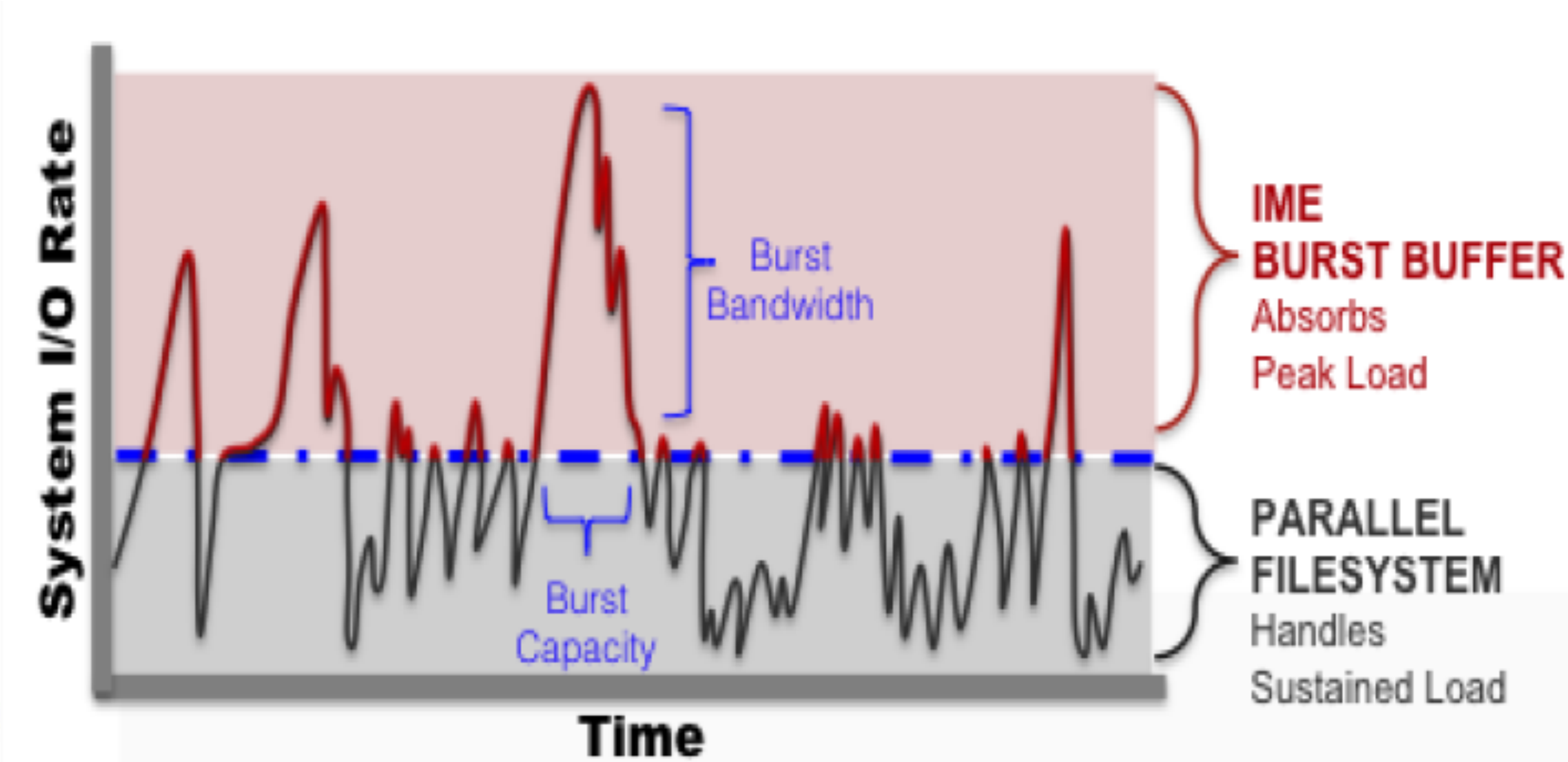
	6TB HDD ($300)	4TB NVMe SSD ($8000)
Capacity	6TB, ~20GB/$	4TB, ~0.5GB/$
BW	150MB/s, ~0.5MB/s/$	3GB/s, ~0.4MB/s/$
IOPs	150/s, ~0.5/$	200,000/s, ~25/$

- **Spinning disk has mechanical limitation in how fast data can be read from disk**
  - SSDs do not have the physical drive components so will always read faster
  - Problem exacerbated for small/random reads
  - But for large files striped over many disks on e.g. Lustre, HDD still performs well.
- **SSDs have limited RWs – the memory cells will wear out over time**
  - This is a real concern for a data-intensive computing center like NERSC.



Why a Burst Buffer?

- **Motivation: Handle spikes in I/O bandwidth requirements**
  - Reduce overall application run time
  - Compute resources are idle during I/O bursts



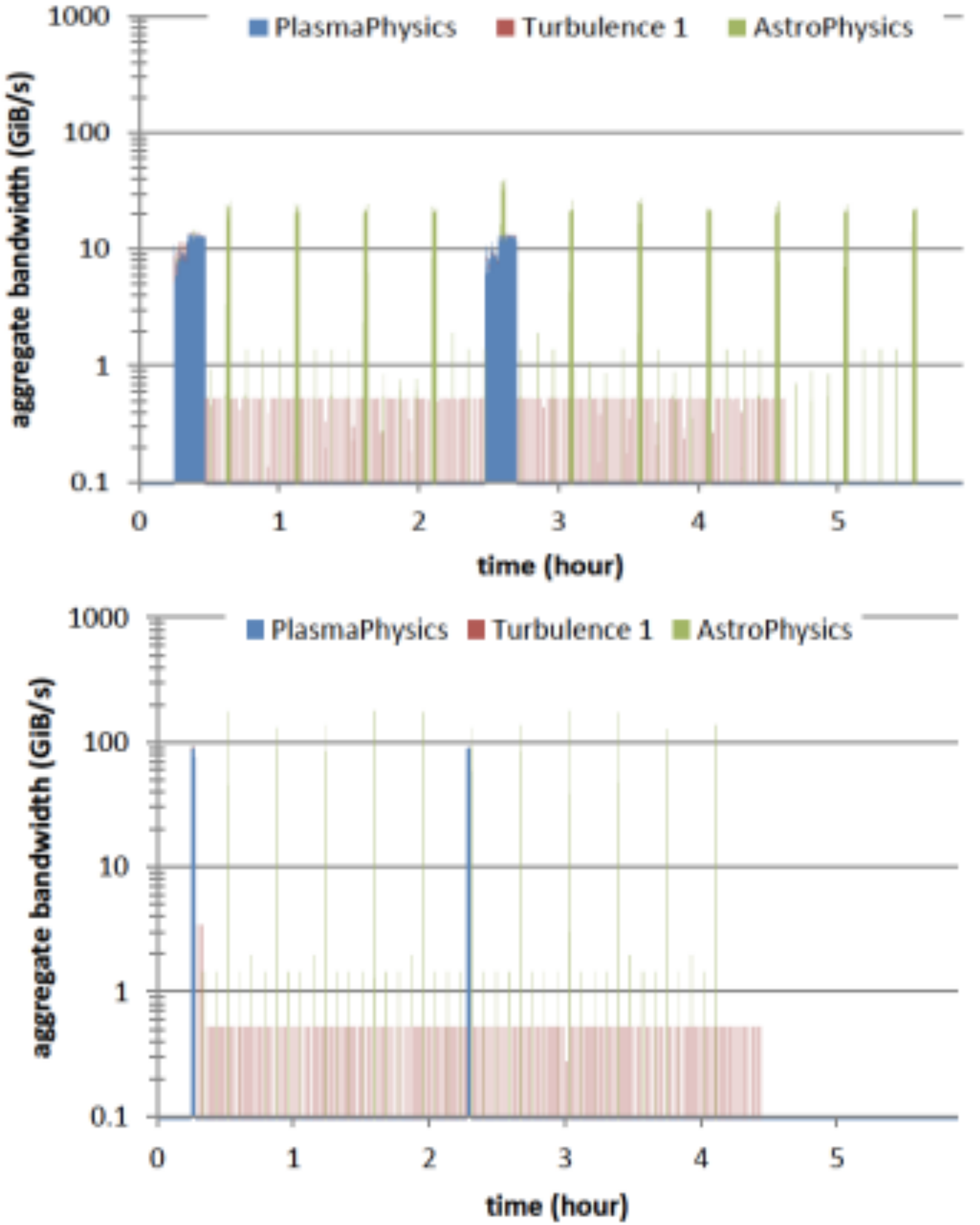
Why a Burst Buffer?

- **Motivation: Handle spikes in I/O bandwidth requirements**
  - Reduce overall application run time
  - Compute resources are idle during I/O bursts
- **Some user applications have challenging I/O patterns**
  - High IOPs, random reads, different concurrency…
- **Cost rationale: Disk-based PFS bandwidth is expensive**
  - Disk capacity is relatively cheap
  - SSD *bandwidth* is relatively cheap

    =>Separate bandwidth and spinning disk
    - Provide high BW without wasting PFS capacity
    - Leverage Cray Aries network speed

Why a Bu

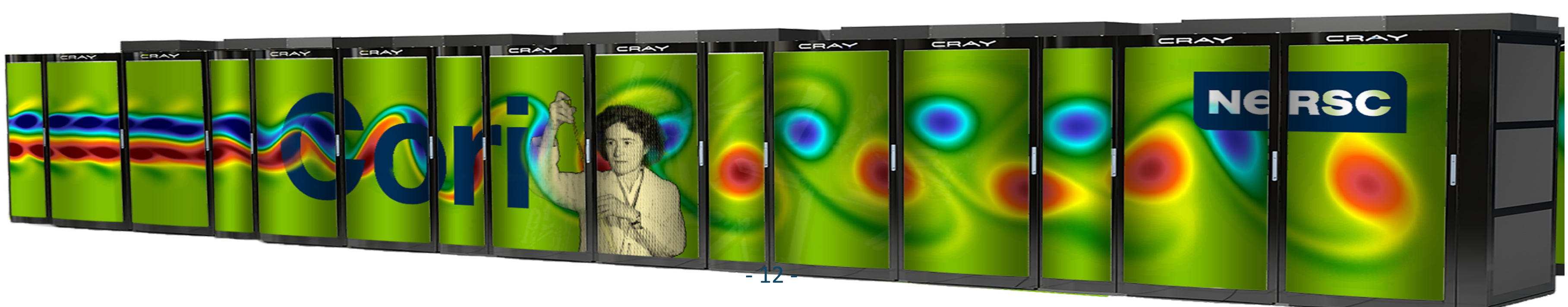
- **Motivatio**
  requireme
  - Reduce
  - Comput
- **Some use** **patterns**
  - High IOP
- **Cost ratio** **xpensive**
  - Disk cap
  - SSD *ban*
    =>Separ
    - Provi
    - Leve



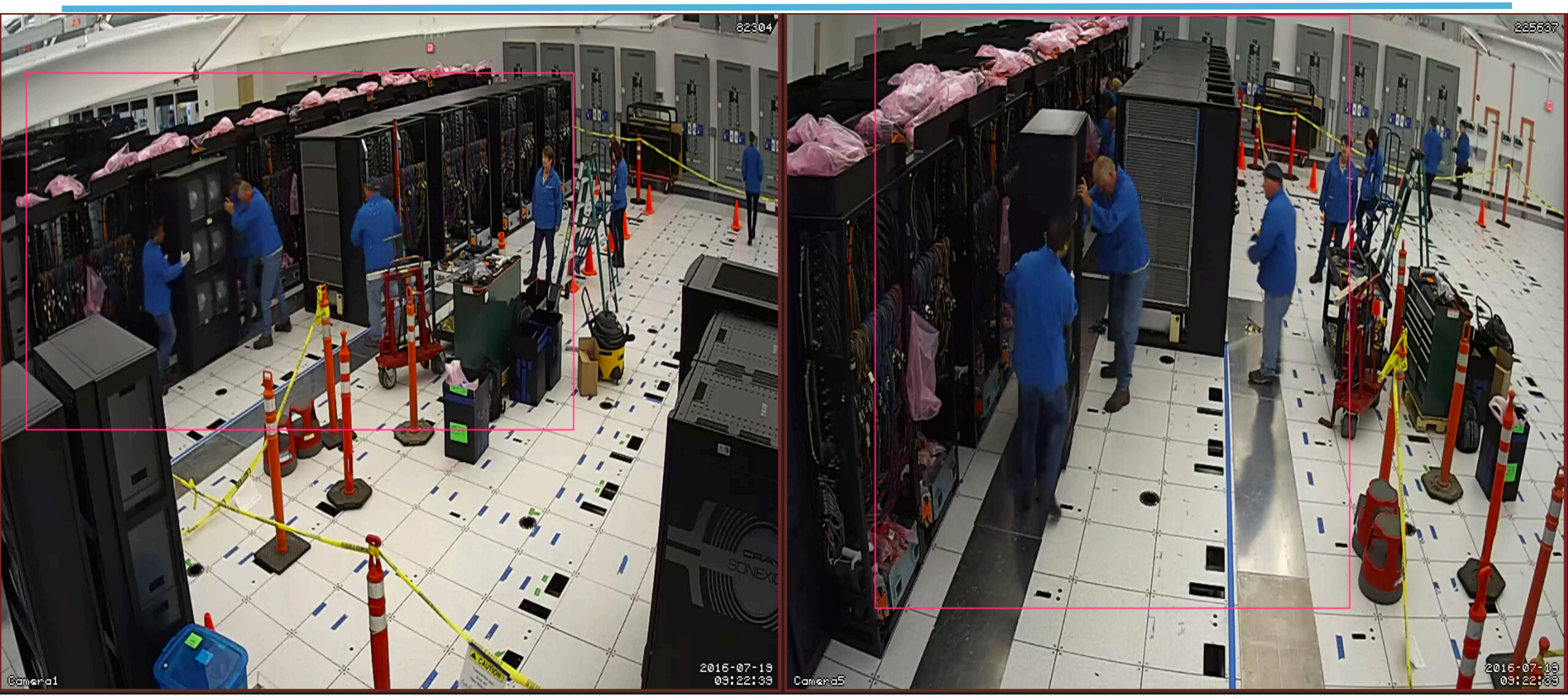
Application perceived I/O rates, with no burst buffer (top), burst buffer (bottom).

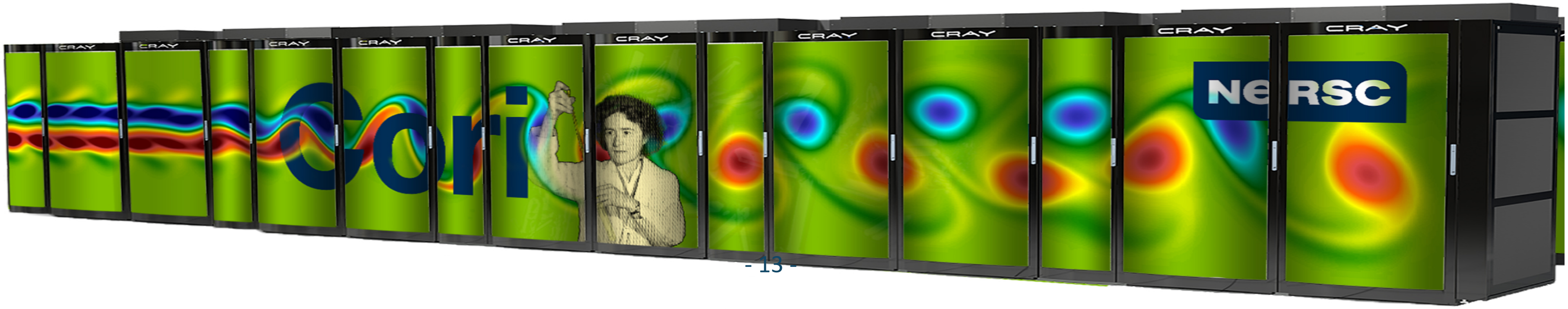
Cori, a Cray XC40 system

- **Cori Phase 1: partition to support data intensive applications**
  - 1630 Intel Haswell nodes
  - Two Haswell processors/node,
    - 16 cores/processor, 128 GB DDR4 /node
- **Cori Phase 2: >9,300 Intel Knights Landing compute nodes**
  - 68 processors/node, 16GB HBM on-package, 96GB DDR4
- **Lustre Filesystem: 27 PB of storage served by 248 OSTs, providing over 700 GB/s peak performance.**
- **Cray Aries high-speed "dragonfly" topology interconnect**
- **1.5PB Burst Buffer…**

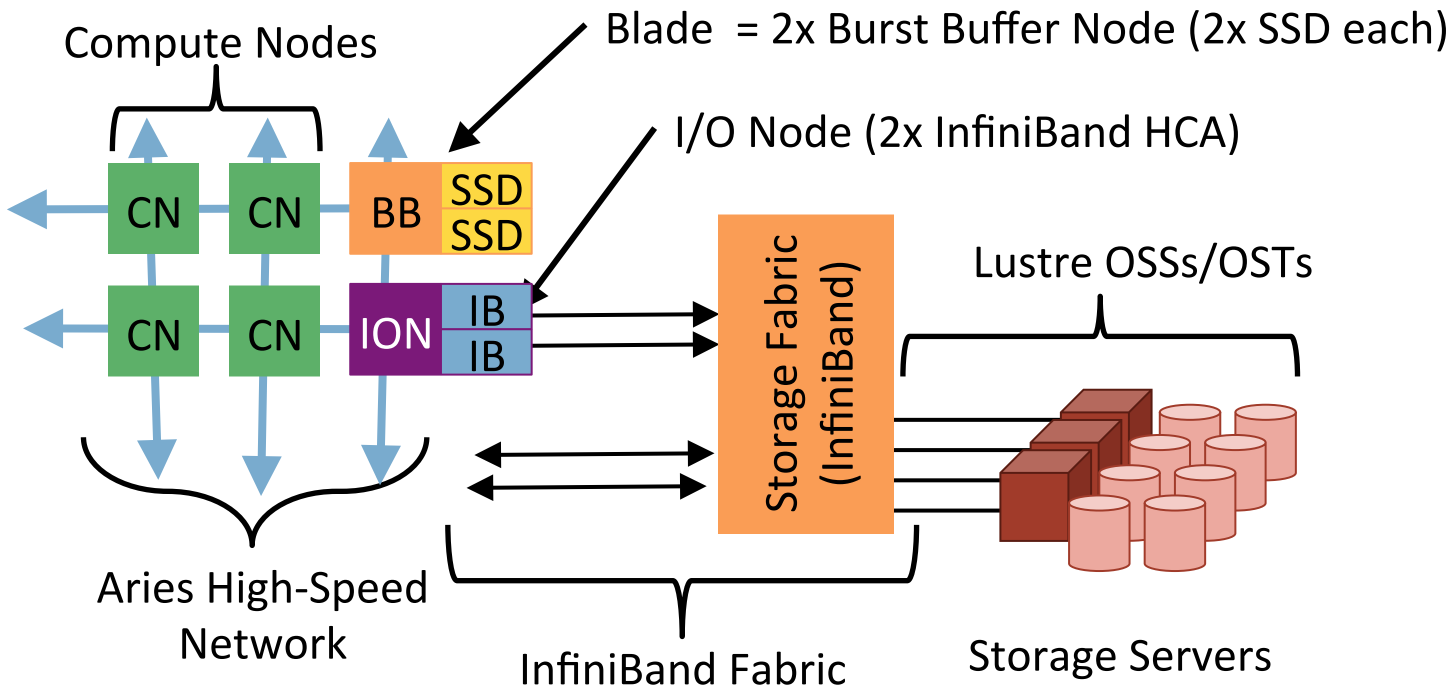


Cori, a Cray XC40 system





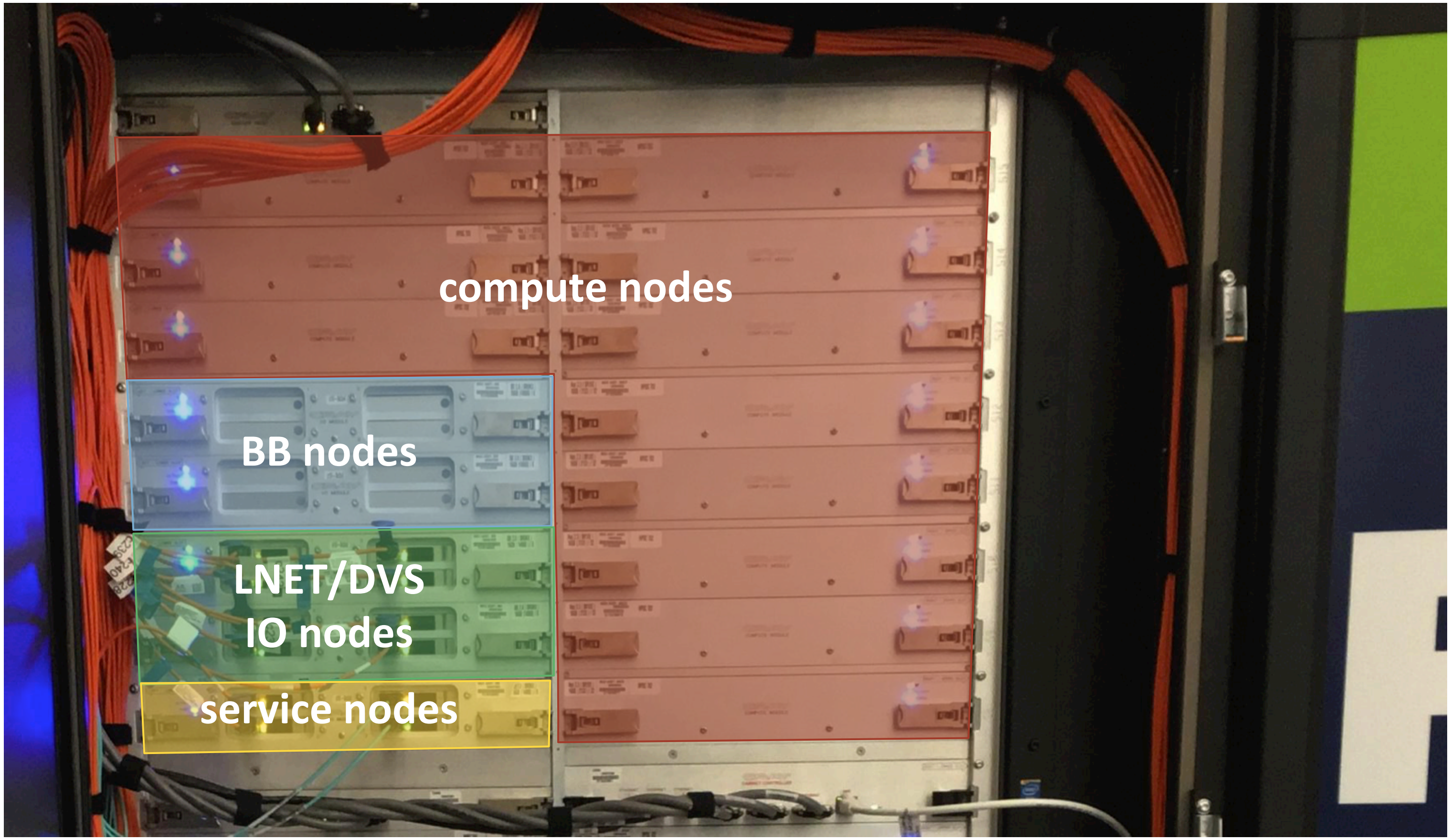
Burst Buffer Architecture



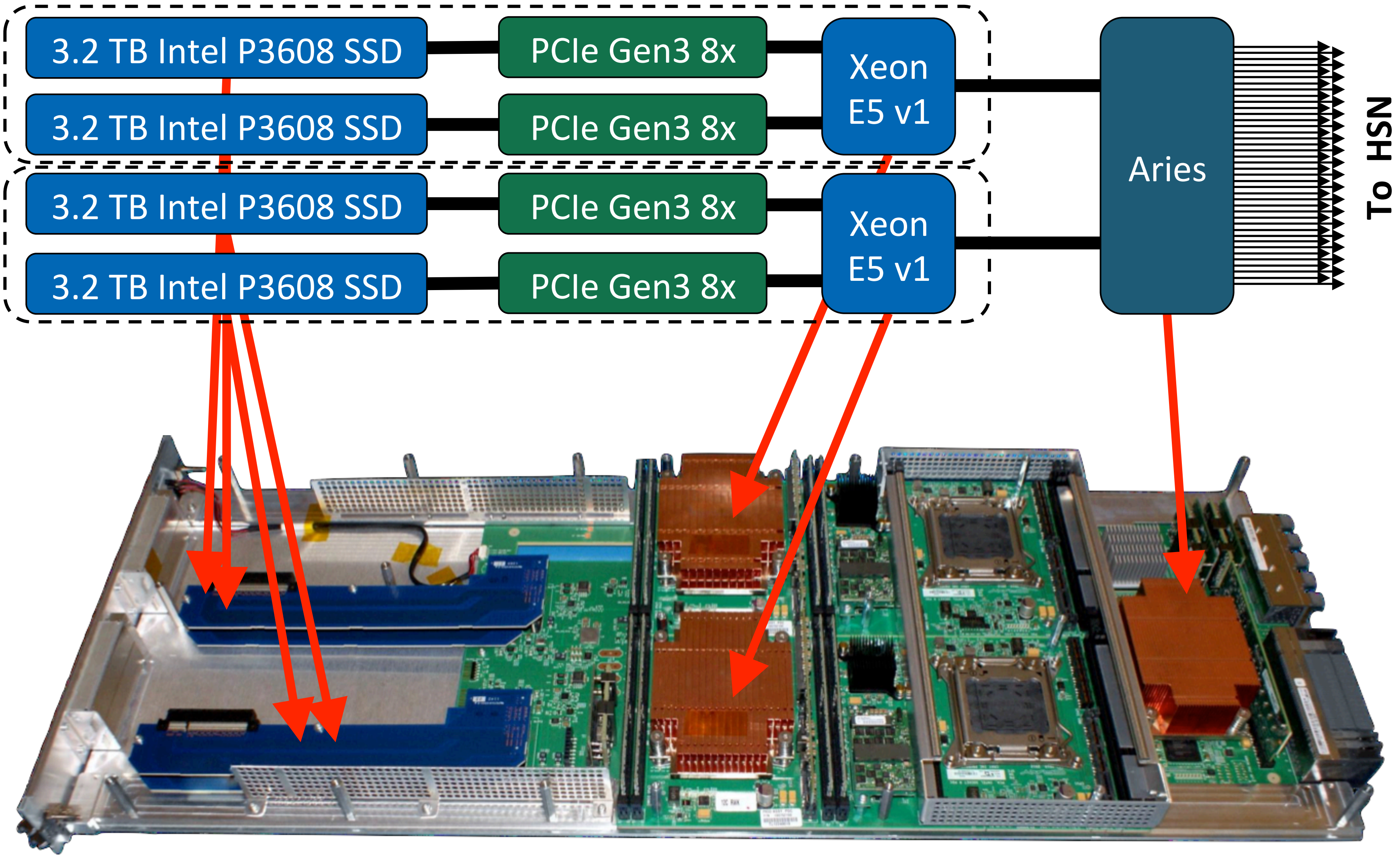
- Cori Stage 1 configuration: 920TB on 144 BB nodes (288 x 3.2 GB SSDs)
- >1.5 PB total in full Cori system

Burst Buffer Architecture Reality

BB nodes scattered throughout HSN fabric
2 BB blades/chassis (12 nodes/cabinet) in Phase I



Burst Buffer Blade = 2xNodes



Why not node-local SSDs?

- **Average >1000 jobs running on Cori at any time**
- **Diverse workload**
  - Many NERSC users are IO-bound
  - Small-scale compute jobs, large-scale IO needs
  - Multi-stage workflows can simultaneously access files on BB.
- **Persistent reservation enables long-term data access without tying up compute nodes**
- **Easier to stream data directly into BB from external experiment**
- ***Configurable BB makes sense for our user load***

Machine Vitals

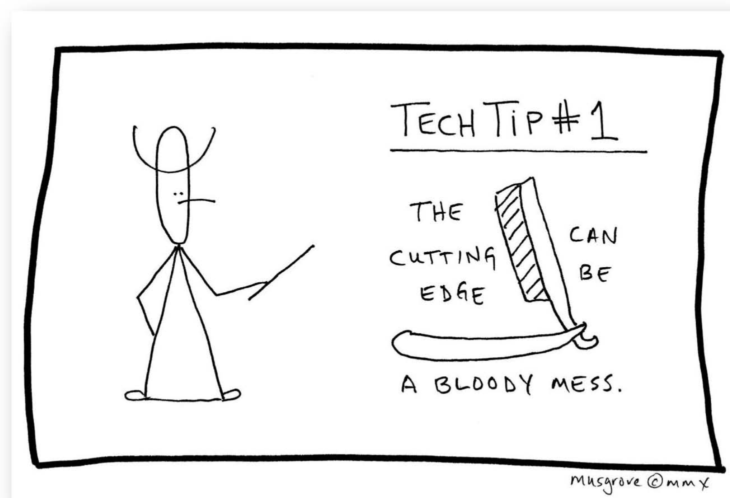
Cori	Cray XC40 **Peak TFlops/s:** (2015)
Peak TFlop/s:	1000
Jobs running:	446
Jobs queued:	2
Cores in use:	46,912 (90%)
Backlog:	0.3 days

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB

New technology needs partnership!

- We're one of the first institutes to deploy a Burst Buffer, and the first to push it beyond the checkpoint/restart use case
- Partnerships with Cray and SchedMD (slurm) are vital to make this work
  - NERSC funds NRE with both Cray and SchedMD
- We're had plenty of teething problems!
- Our early users have been major debuggers of the software.

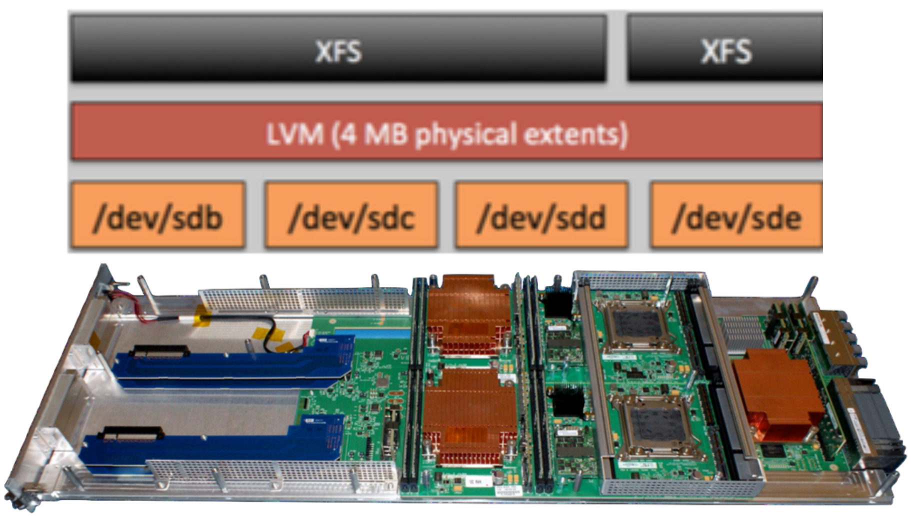


Cray DataWarp implementation

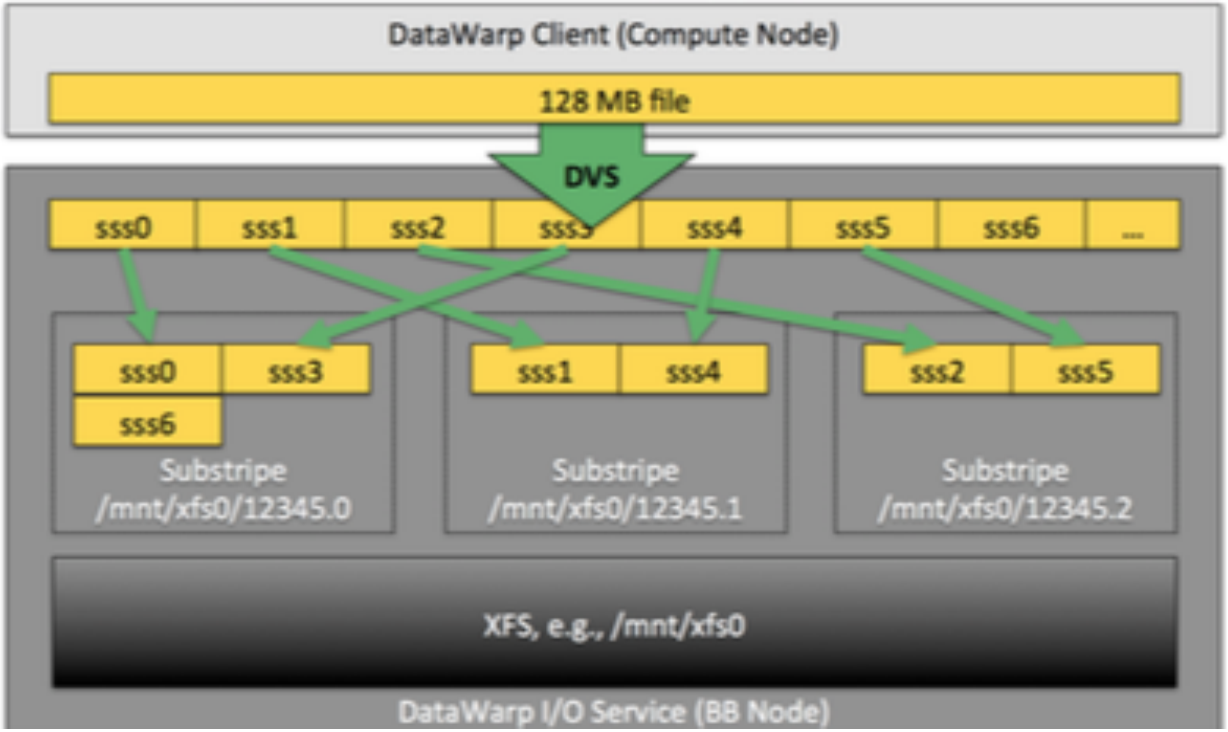
- **High performance SSDs in service nodes, directly attached to Aries network**
  - Software creates pool of available storage
  - Allocate portions of this pool to users per-job, or in a persistent reservation
  - Users see a ***POSIX filesystem created for their use***
- **Potential performance benefits for many reasons:**
  - Underlying storage media is fast
  - Placed inside high-performance network
  - Namespace is per job or workflow – limited metadata load
  - Asynchronous transfer to PFS
  - Users have access to 100s of TBs from one or many compute nodes: flexible configuration.

Filesystem layers

- **Logical Volume Manger (LVM)** groups the 4 SSDs into one block device.
- An **XFS** file system is created for every Burst Buffer allocation
  – Per-job “scratch”, or persisitent reservation.
- **DataWarp File System (DWFS)**: stacked file system providing the namespaces.
- **Cray Data Virtualization Service (DVS)**: mediates communication between DWFS and the compute nodes.



Filesystem layers



- One 128MB file ends up as (configurable) 8MB chunks, laid out across the three (configurable) substripes on the Burst Buffer node.

Integrated with SLURM WLM – easiest user interface

```
#!/bin/bash
#SBATCH -p debug -N 1 -t 00:10:00

#DW jobdw capacity=200GB  access_mode=striped type=scratch

#DW stage_in source=/lustre/inputs  destination=$DW_JOB_STRIPED/inputs type=directory
#DW stage_in source=/lustre/file.dat   destination=$DW_JOB_STRIPED/ type=file

#DW stage_out source=$DW_JOB_STRIPED/outputs   destination=/lustre/outputs type=directory

srun my.x --indir=$DW_JOB_STRIPED/inputs --infile=$DW_JOB_STRIPED/file.dat --outdir=$DW_JOB_STRIPED/outputs
```

- **Example illustrates**
  - –Duration of allocation 'type=scratch' is just for compute job
  - –'access_mode=striped' – visible to all compute nodes and can be striped across multiple BB nodes (alternative is 'private')
    - Actual distribution across BB Nodes in units of granularity (currently 200 GB so 1000 GB would normally be placed on 5 BB nodes)
  - –Data can be staged in and out

Benchmark Performance

- **Burst Buffer is exceeding (nearly all) benchmark performance targets**
  - MPIO shared file write has since been improved (but we haven't re-run the benchmark yet)
  - Out-performs Lustre (Lustre also exceeds requirements)

	140 Burst Buffer Nodes : 1120 Compute Nodes; 4 processes/node					
	IOR Posix FPP		**IOR MPIO Shared File**		**IOPS**	
	Read	**Write**	**Read**	**Write**	**Read**	**Write**
Best Measured	**905 GB/s**	**873 GB/s**	**803 GB/s**	**351 GB/s**	**12.6 M**	**12.5 M**
Lustre (peak)	708 GB/s	751 GB/s	573 GB/s	223GB/s	-	-

*Bandwidth tests: *8 GB block-size 1MB transfers*
IOPS tests: 1M blocks 4k transfer

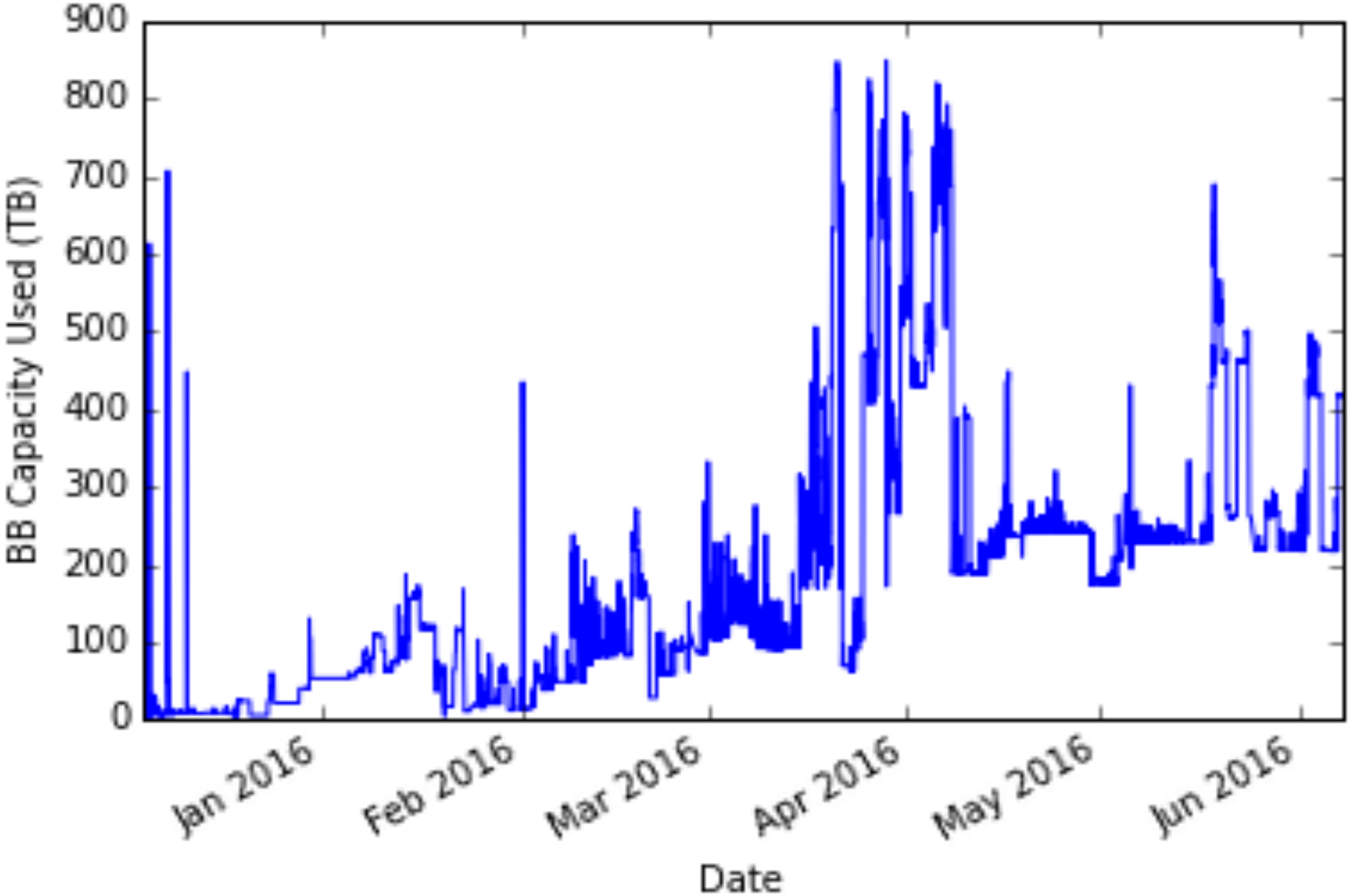
Burst Buffer Early User Program

- **NERSC has most diverse user base of all DOE computing facilities: Over 6500 users on more than 700 projects, running 700+ codes**
- August: solicited proposals for BB Early Users program.
- Great interest from the community, ~30 proposals received.
- Selection criteria include:
  - Scientific merit; Computational challenges; Cover range of BB data features; Cover range of DoE Science Offices.
- Support ~10 applications actively
  - some applications already had LDRD funding at LBNL, and existing support from NERSC staff.
- ~20 applications not supported by NERSC staff, but have early access to Cori P1 and the BB.

User Experience ≠ benchmark

- **Significant number of major software bugs continue to impact user experience**
  – Most have been quickly patched by Cray
- **Minor bugs/quirks cause some frustrations**
  – E.g. formatting requirements,
  – Also quickly patched by Cray
- **Few users saw OOTB improvement in IO**
  – Most saw (see) far better performance on Lustre
  – Significant effort required to get good performance out of existing code

Burst Buffer Occupation



~50 active users, not general access

Burst Buffer Use-cases

Burst Buffer User Case	Example Early Users
IO Bandwidth: Reads/ Writes	• **Nyx/BoxLib astro sims** • VPIC IO plasma sims
Data-intensive Experimental Science - "Challenging" IO pattern, eg. high IOPs	• **ATLAS HEP experiment** • TomoPy for ALS and APS • Genome assembly codes
Workflow coupling and visualization: in transit / in-situ analysis	• **ChomboCrunch & VisIt carbon sequestration simulation** • Climate simulation/visualization • Electron cryo-microscopy image assembly/visualization
Staging experimental data	• **ATLAS HEP experiment** • ALS SPOT Suite • Tractor astronomy image analysis

(note: no out-of-core use cases applied)

Many others projects listed in backup slide

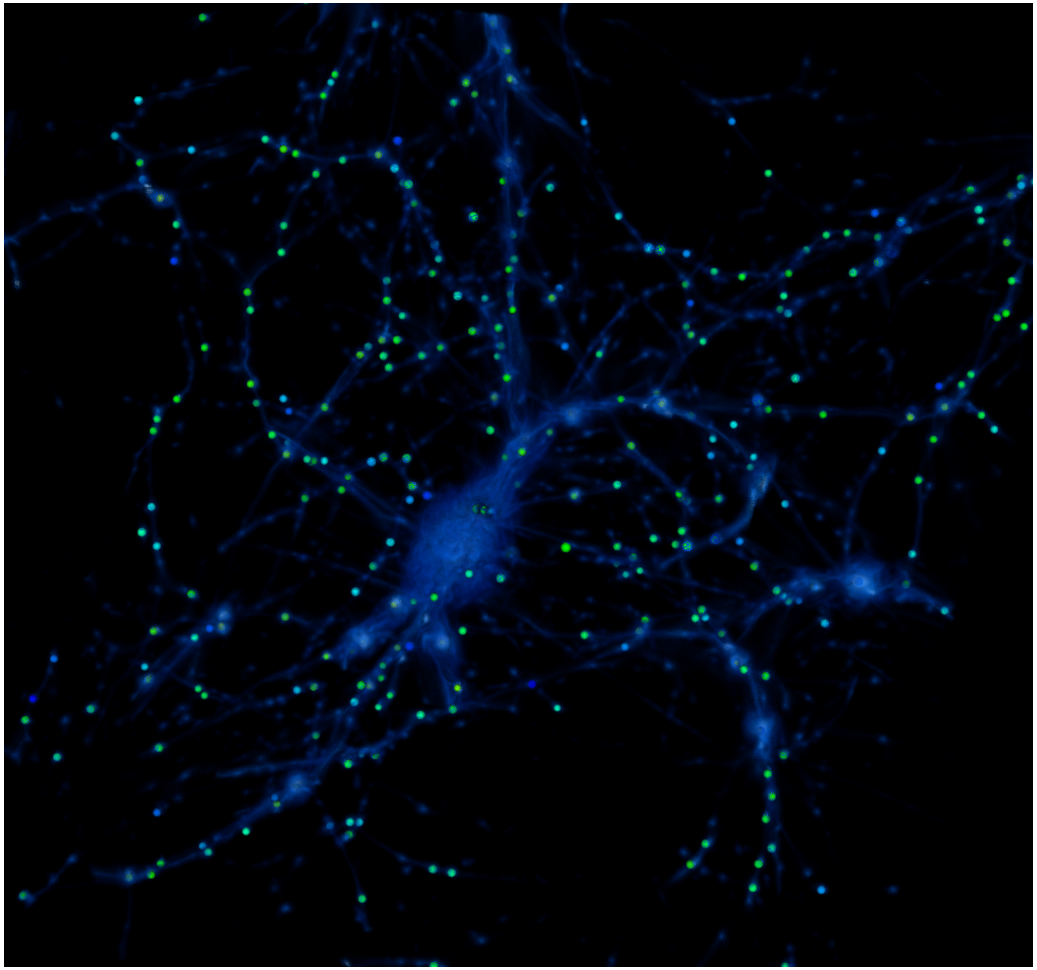
I/O Performance: Read/Write

- **Classic "checkpoint" use case also applies to our data-intensive users writing out large simulation data files**
- **To maximise BB BW, we need to keep it busy:**
  - Need >4 processes writing to a BB node
  - Need large transfer sizes
- **Use cases that fit this I/O pattern (or can adapt to it) saw excellent performance compared to Lustre**

I/O R/W use case: Nyx/Boxlib

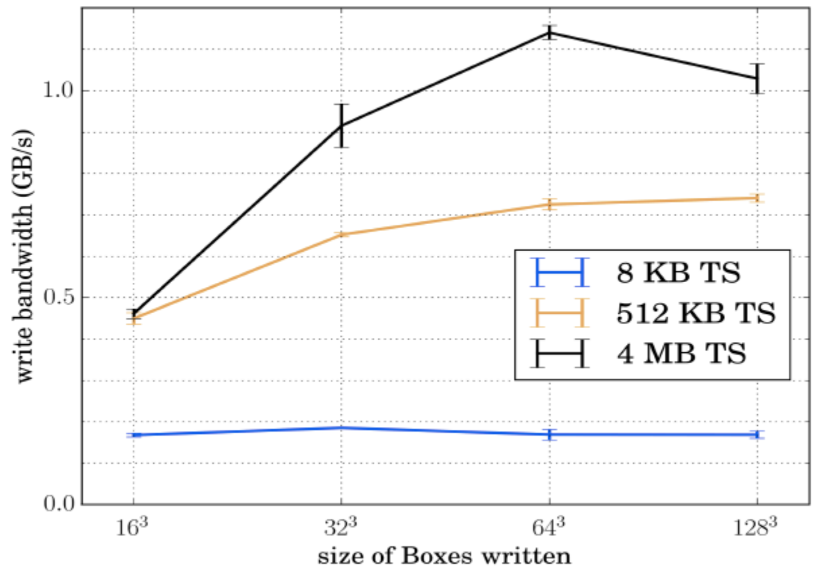
Brian Friesen, Ann Almgren

- Nyx cosmological simulation code based on a widely-used adaptive mesh refinement (AMR) library, BoxLib
- Large data files ("plotfiles") written at certain time steps; checkpoint files too
- I/O time consumes a significant fraction of run time



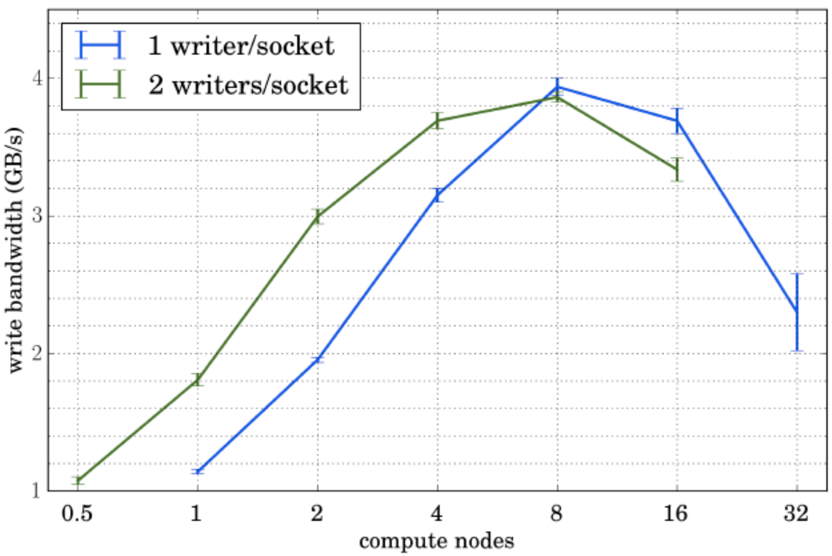
I/O R/W use case: Nyx/Boxlib

- **Need larger transfer size for good performance**



I/O R/W use case: Nyx/Boxlib

- **Need larger transfer size for good performance**
- **Need >16 MPI writers per BB node for performance**

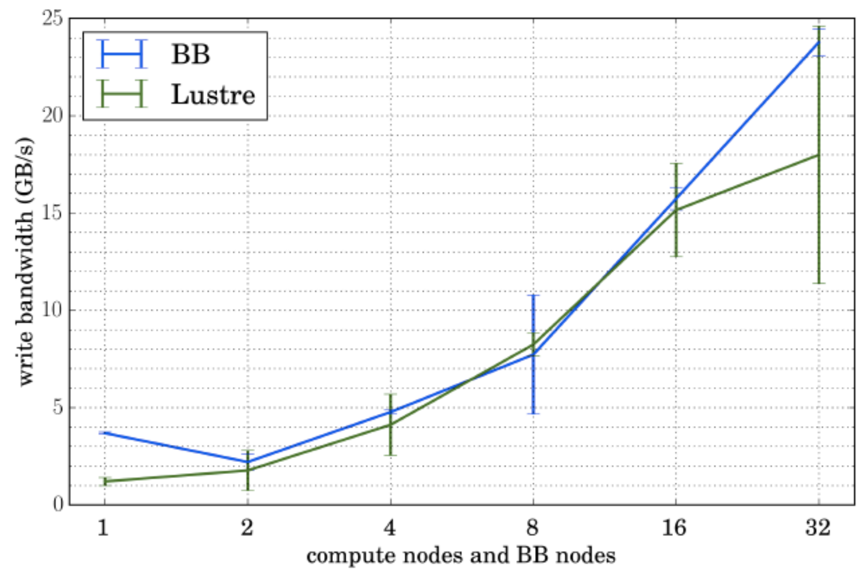


I/O R/W use case: Nyx/Boxlib

- **Need larger transfer size for good performance**
- **Need >16 MPI writers per BB node for performance**
- **BB performance scales up as you increase # BB nodes in allocation**

I/O R/W use case: Nyx/Boxlib

- **Need larger transfer size for good performance**
- **Need >16 MPI writers per BB node for performance**
- **BB performance scales up as you increase # BB nodes in allocation**
- ***BB performance matches Lustre***

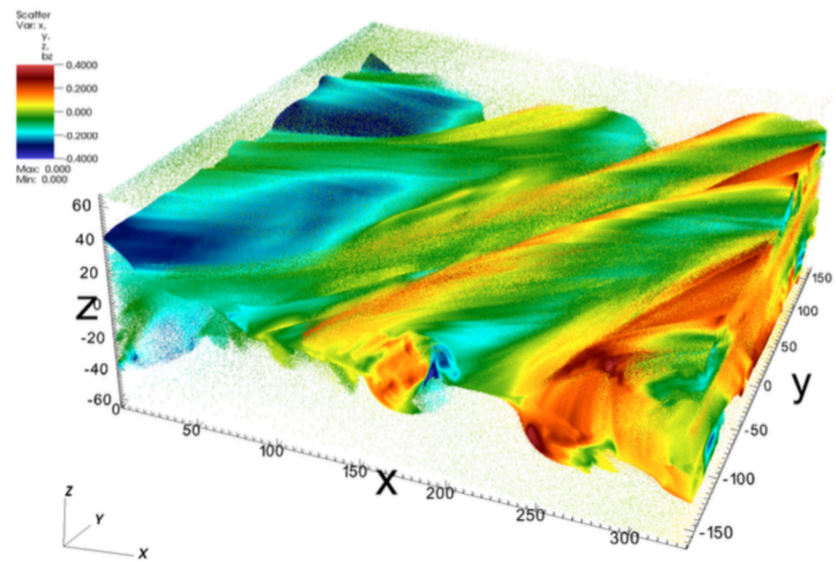


- **Note that this does not necessarily correspond to optimal Nyx compute configuration!**

I/O R/W use case: VPIC I/O

Matt Bryson, Suren Byna, Glenn K. Lockwood

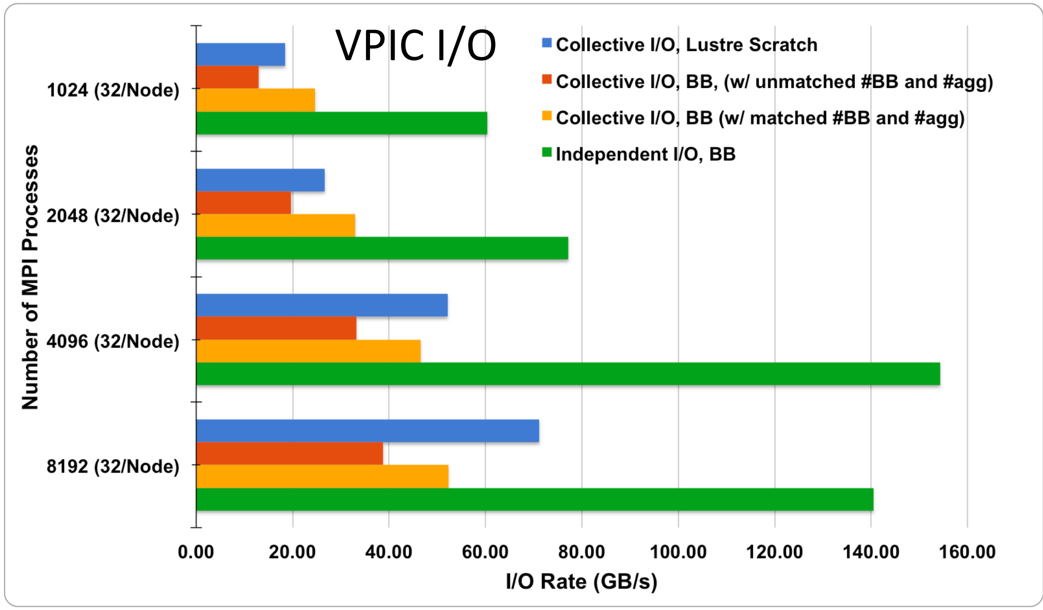
- Plasma physics simulation
- Shared file I/O using HDF5
- Can be large amount of data e.g. magnetic reconnection with two trillion particles – 32-40 TB per time step
- Write out each time step to Burst Buffer with asynchronous copy to PFS
- Also potential for in-transit visualization



- 34 -

VPIC I/O: MPI-IO Collective

- Using 65 Burst Buffer nodes 'unmatched' with collective MPI aggregators – poor performance
- 64 BB nodes – 'matched' – significantly better
  - Comparable with Lustre
- Independent I/O performs 4x better
- Profile with Darshan and VPIC-like IOR run confirms MPI collective overhead



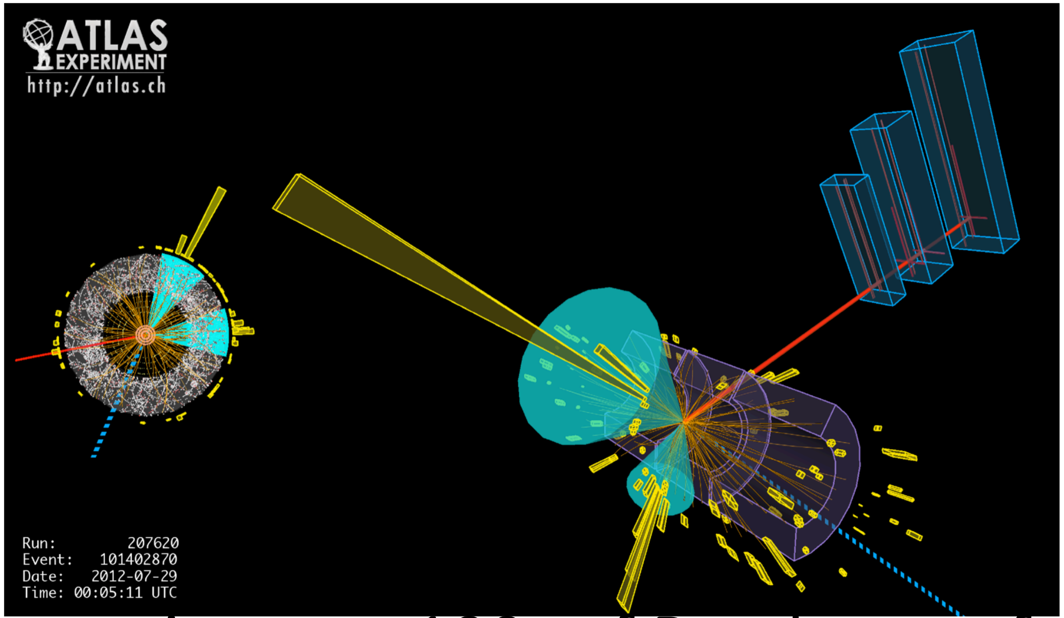
IOR based modeling of I/O pattern:

API	Mean B/W (GB/s)
HDF5	14.7
MPIIO	15.4
POSIX	66.5

Challenging I/O patterns

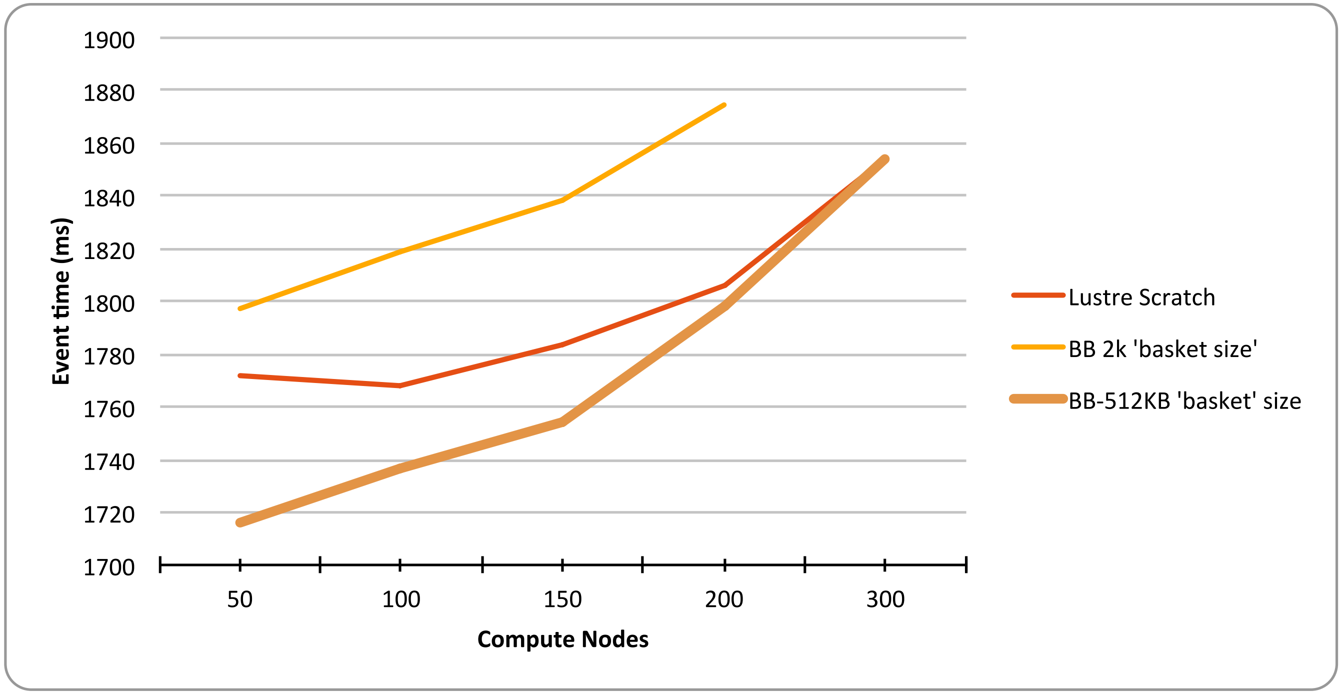
- **Benchmarks show promising results**
  - 12M IOP/s!
- **Reality more complex**
- **Lack of client-side caching significantly impacts performance compared to Lustre**
- **Applications tuned to use larger transfer sizes etc saw better performance**
  - *Make them more like checkpoint use case*
- **DVS client-side caching and metadata improvements will help (coming later this year from Cray)**

Challenging IO use case: ATLAS/Yoda



- ATLAS LHC experiment – 100s of Petabytes of data processed worldwide - but little use of 'HPC' machines
- 'Yoda' packages ATLAS payloads for HPC
  - Used in production but running least I/O intensive simulation
  - Use Burst Buffer to run I/O intensive analysis

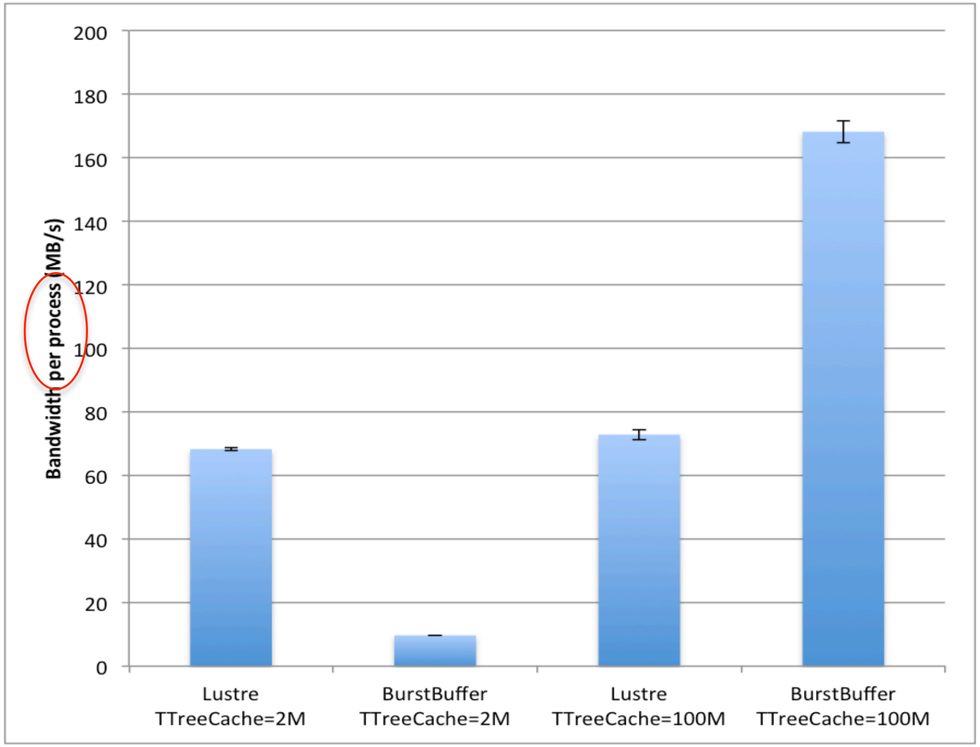
Challenging IO use case: ATLAS/Yoda



- Initial scaling on BB poor
- Increase ROOT 'basket size' from 2k to 512k to increase transaction size
- Keep log files on Lustre
- Then scales to >300 nodes
- But this is not most I/O intensive payload…

Challenging IO use case: ATLAS data

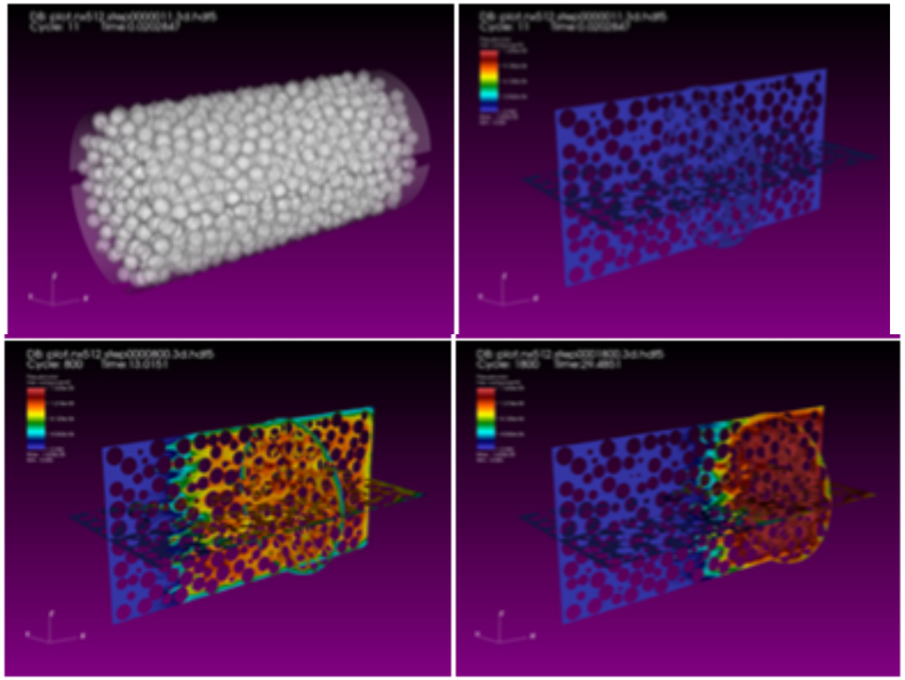
- Initial study of I/O intensive data processing
- Reading 475 GB dataset in custom ROOT format
- 32 forked processes per node, FPP R/W
- Initial result: BB performs poorly compared to Lustre.



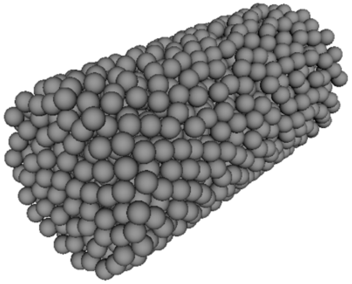
- Increase application memory cache to 100 M
- Less reads – > 17x performance boost on BB

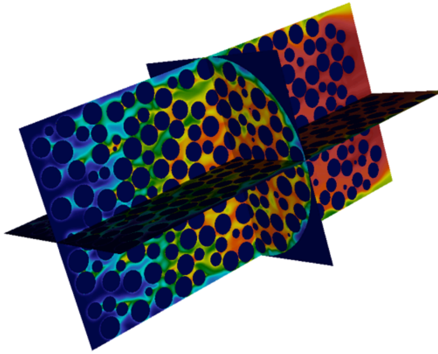
Workflow coupling and visualization

- **Success story: Burst Buffer can enable new workflows previously difficult to orchestrate using Lustre alone**



Workflows Use Case: ChomboCrunch + VisIT

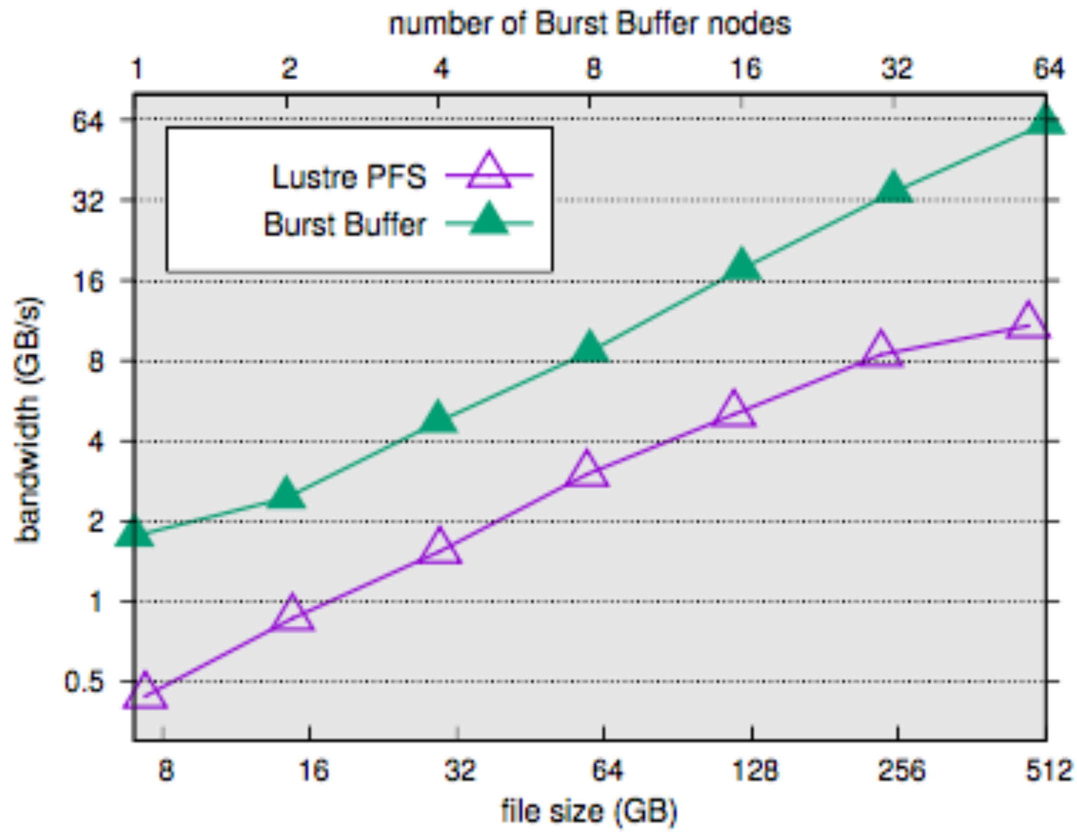




- **ChomboCrunch simulates pore-scale reactive transport processes associated with carbon sequestration**
  - Flow of liquids through ground layers
  - All MPI ranks write to single shared HDF5 '.plt' file.
  - Higher resolution -> more accurate simulation -> more data output (O(100TB))
- **VisIT – visualisation and analysis tool for scientific data**
  - Reads '.plt' files produces '.png' for encoding into movie
- **Move from using Lustre to *store* intermediate files**

Scaling

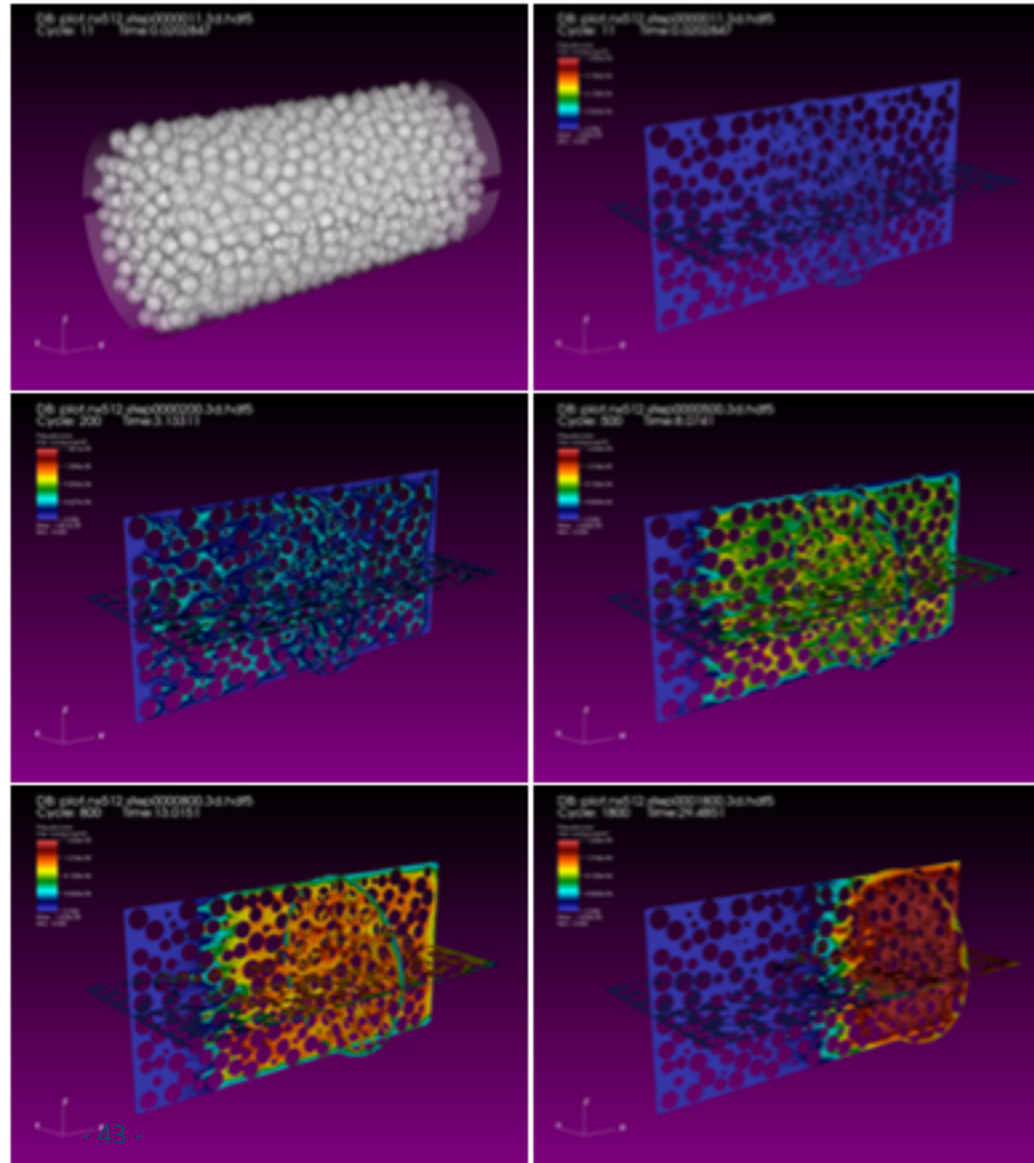
- **Burst Buffer significantly out-performs Lustre for this application at all resolution levels**
  – Did not require any additional tuning!
- **Bandwidth achieved is around a quarter of peak, scales well.**



Compute node/BB node scaled: 16/1 to 1024/ 64
Lustre results used a 1MB stripe size and a stripe count of 72 OSTs

In-transit Movie

- Simulation ran on 8192 cores over 256 nodes with 8 further nodes used for VisIt.
- 140 BB nodes:
  - 90.7GB/s obtained
  - (840 GB/s theoretical)
- **A coupled science workflow using the Burst Buffer**



U.S. DEPARTMENT OF ENERGY | Office of Science

In-transit Movie

A coupled science workflow using the Burst Buffer

Summary: User Experience so far

- **Writing large files (with large block I/O) is fast (checkpointing use case)**
- **Reading/Writing small files (or small I/O transfers) is problematic in some cases**
  - Generally in many cases our BB performance is worse than our Lustre filesystem (which is high-performance).
  - Client-side caching helps Lustre performance
- **Still some system instabilities**
- **Initial enthusiasm from users somewhat diminished, but not extinguished!**
  - Continue to get requests to access BB.

Lessons Learned

- **Not seen *immediate* payoff for any user code**.
  - Despite good benchmark performance
- **Challenging I/O patterns do see some benefit**
  - More tuning required – not even close to peak BW
- **MPI-IO with Burst Buffers will require further tuning to perform well**.
  - ~5 years of work went into MPI-IO for Lustre
  - Hints that DWFS/MPI-IO transfers are not in tune
- ***Tuning of transfer size and number of parallel writers is needed with the Burst Buffer*, more so than with Lustre**.

Conclusions

- **Cori has one of the first fully functional Burst Buffers in the world**
  - And the first to be tested beyond checkpoint/restart
- **Users are enthusiastic about new memory hierarchy!**
- **Burst Buffer has demonstrable utility beyond checkpoint/restart use case**
- **Very promising IO accelerator, but early stage of development**
  - Benchmarks good, user experience mixed…
- **Early User program excellent debugger of new hardware**

NERSC

Thankyou

Use Cases by BB feature

Application	I/O bandwidth: reads	I/O bandwidth: writes (checkpointing)	High IOPs	Workflow coupling	In-situ / in-transit analysis and visualization	Staging intermediate files/ pre-loading data
Nyx/Boxlib		X		X	X	
Phoenix 3D		X		X		X
Chomo/Crunch + Visit		X		X	X	
Sigma/UniFam/Sipros	X	X	X			X
XGC1	X	X				X
PSANA				X	X	X
ALICE	X					
Tractor			X	X		X
VPIC/IO					X	X
YODA			X			X
ALS SPOT/TomoPy	X			X	X	X
kitware				X	X	

Use Cases by BB feature

Application	I/O bandwidth: reads	I/O bandwidth: writes (checkpointing)	High IOPs	Workflow coupling	In-situ / in-transit analysis and visualization	Staging intermediate files/ pre-loading data
Electron cryo-microscopy						X
htslib						X
Falcon	X	X				
Ray/HipMer	X	X	X			X
CESM	X	X				
ACME/UV-CDAT					X	X
GVR		X				
XRootD				X		X
OpenSpeedShop	X	X				
DL-POLY		X				
CP2K		X				
ATLAS	X		X			X